Proceedings of the Thirty-Sixth International Conference on Automated Planning and Scheduling (ICAPS 2026)

Boltzmann-based Exploration for Robust Decentralized Multi-Agent Planning

Nhat Nguyen[1,2], **Duong Nguyen**[1], **Gianluca Rizzo**[3, 4], **Hung Nguyen**[1]
[1]Adelaide University, Australia
[2]Università di Foggia, Italy
[3]HES-SO Valais, Switzerland
[4]Università di Torino, Italy
nhatdaoanh.nguyen@adelaide.edu.au, duong.nguyen@adelaide.edu.au, gianluca.rizzo@hevs.ch, hung.nguyen@adelaide.edu.au

Abstract

Decentralized Monte Carlo Tree Search (Dec-MCTS) is widely used for cooperative multi-agent planning but struggles in sparse or skewed reward environments. We introduce Coordinated Boltzmann MCTS (CB-MCTS), which replaces deterministic UCT with a stochastic Boltzmann policy and a decaying entropy bonus for sustained yet focused exploration. While Boltzmann exploration has been studied in single-agent MCTS, applying it in multi-agent systems poses unique challenges. CB-MCTS is the first to address this. We analyze CB-MCTS in the simple-regret setting and show in simulations that it outperforms Dec-MCTS in deceptive scenarios and remains competitive on standard benchmarks, providing a robust solution for multi-agent planning.

Introduction

Decentralized Monte Carlo Tree Search (Dec-MCTS) is an increasingly popular paradigm for cooperative multi-agent planning (Best et al. 2019; Li et al. 2019; Nguyen et al. 2024b). Its anytime performance, domain-agnostic design, and online replanning capabilities make it well-suited for applications requiring scalability, fast response, and coordination across distributed agents, such as information gathering, precision farming, and networked robotics (Claes et al. 2017; Sukkar et al. 2019; Nguyen et al. 2024a).

Current Dec-MCTS algorithms rely on the Upper Confidence Bound applied to Trees (UCT) and its variants (Kocsis, Szepesvári, and Willemson 2006) to guide the search process. UCT selects actions according to the principle of *optimism in the face of uncertainty*, prioritizing branches with high empirical rewards. While this mechanism is effective when rewards are smooth or moderately stochastic (Munos et al. 2014), its efficiency degrades in skewed, sparse, or deceptive reward landscapes. In such cases, early high-reward samples can mislead the search, causing the algorithm to overcommit to suboptimal branches while neglecting deeper paths that lead to higher rewards (Coquelin and Munos 2007; Ramanujan and Selman 2011; James, Konidaris, and Rosman 2017). Although extensively studied in single-agent MCTS, the implications for decentralized multi-agent planning, where coordination amplifies the problem, remain largely unexamined.

This paper provides the first simple regret analysis of Dec-MCTS in deceptive multi-agent trees. We then introduce *Coordinated Boltzmann Monte Carlo Tree Search* (CB-MCTS), a distributed algorithm that addresses these limitations. CB-MCTS replaces the deterministic UCT selection with a stochastic Boltzmann policy and incorporates a decaying entropy-based bonus to sustain exploration while progressively focusing on high-value actions. Coordination among agents is achieved through a marginal contribution function that aligns each agent's local decisions with the global objective, mitigating the variance introduced by simultaneous actions. This approach enables the search to explore deceptive or initially suboptimal regions effectively, improving convergence to globally optimal strategies.

While Boltzmann exploration has been applied in single-agent MCTS (Cesa-Bianchi et al. 2017; Painter et al. 2023), CB-MCTS is, to our knowledge, the first to adapt it to multi-agent planning. We show theoretically that CB-MCTS achieves exponentially faster decay of simple regret than D-UCT-based Dec-MCTS in deceptive trees. Extensive simulations demonstrate that CB-MCTS matches state-of-the-art methods on standard benchmarks while significantly outperforming them in scenarios with skewed or sparse reward distribution. Overall, CB-MCTS offers a robust and adaptable framework for multi-agent planning problems across smooth to sparse reward environments.

Problem Statement

We consider a cooperative multi-agent planning problem with N agents in a shared environment modeled as an undirected graph $G = (V, E)$, where vertices are states and edges are actions. Each agent n selects a valid action sequence $a^n \in \mathcal{A}^n$, subject to a cost budget $b(a^n) \leq B^n$. The global objective $g(a)$ depends on the joint action $a = (a^1, \ldots, a^N)$, and the goal is to maximize $g(a)$ within a planning budget T. Dec-MCTS addresses this by letting each agent build its own search tree using repeated simulations. A trajectory from the root corresponds to a candidate action sequence, and nodes are selected using Discounted UCT (D-UCT), which weights empirical rewards by a discount factor γ.

While cumulative regret is standard in MCTS evaluation, in multi-agent planning with finite planning budgets, only the executed actions contribute to real-world outcomes.

Hence, *simple regret* $r_T = \mu^* - \mu_{J_T}$, the expected loss of executing the recommended action J_T after T planning iterations, is a more relevant metric (Tolpin and Shimony 2012). While UCT is guaranteed to converge to the optimal trajectory in the limit (Kocsis, Szepesvári, and Willemson 2006), this convergence can be extremely slow in sparse, skewed, or deceptive reward structures, such as the classical D-chain tree (Coquelin and Munos 2007; James, Konidaris, and Rosman 2017). We show that this D-chain pathology extends to Dec-MCTS with D-UCT, where the difficulties are magnified, particularly for multi-agent coordination.

Definition 1 *The multi-agent D-chain problem is an M-ary tree of depth D. At depth $d < D$, action 1 progresses to the next level; all others terminate with reward $(D - d)/D$. At depth D, action 1 gives a reward of 1; others give 0. Agents initially select non-progressing actions, and overlapping action sequences among agents do not yield extra reward.*

Lemma 1 *For a fixed γ, there exists a value D such that Dec-MCTS with D-UCT fails to identify the optimal action sequence in the D-Chain problem.*

As D-UCT is designed to minimize cumulative regret, it is unsuitable for environments that require extensive exploration (i.e., minimizing simple regret) since both regrets cannot be minimized simultaneously (Bubeck, Munos, and Stoltz 2011). We formally bound the simple regret of Dec-MCTS with D-UCT as follows:

Theorem 1 *The simple regret of Dec-MCTS with D-UCT is bounded by $\mathbb{E}[r_T] \leq C \exp\left(-k\sqrt{T}\log T\right)$ for some constants $C, k > 0$.*

The proofs of Lemma 1 and Theorem 1 are in the extended version (Nguyen et al. 2026).

Coordinated Boltzmann MCTS

Distributed CB-MCTS with Discounted Backup

We propose Coordinated Boltzmann Monte Carlo Tree Search (CB-MCTS), a distributed algorithm for cooperative multi-agent planning. Each agent n independently runs CB-MCTS over its search tree $\mathcal{T}^n$, where nodes represent states and edges represent actions. A root-to-leaf branch encodes a feasible action sequence. All agents aim to maximize the global utility g through decentralized coordination.

To coordinate without centralization, each agent maintains a compressed representation of its tree consisting of (i) a subset $\hat{\mathcal{A}}^n$ of high-value rollouts and (ii) a probability mass function p^n over these rollouts. The subset $\hat{\mathcal{A}}^n$ is obtained by selecting leaf nodes with the highest discounted empirical returns every c iterations. The probabilities p^n are updated via a decentralized gradient-based consensus protocol (Best et al. 2019), enabling each agent to form beliefs about others' future trajectories without exchanging full trees.

The tree $\mathcal{T}^n$ is grown iteratively using the standard 4-step MCTS process (Kocsis, Szepesvári, and Willemson 2006). During *selection*, the algorithm recursively chooses a child using the stochastic Boltzmann policy, stopping when encountering an unvisited child or reaching the planning horizon. The chosen unvisited child is then *expanded*. A *rollout* phase follows, where random actions are sampled until the horizon is reached.

Algorithm 1: Overview of CB-MCTS for agent n

Require: $g, T, c, B, \hat{\mathcal{A}}^{-n}, p^{-n}$
Ensure: best action sequence a^n for agent n

$\mathcal{T}^n \leftarrow$ Initialize the search tree
$t \leftarrow 0$
while $t < T$ **do**
  if $t \bmod c == 0$ **then**
    $\hat{\mathcal{A}}^n \leftarrow$ Tree Compression $(\mathcal{T}^n)$
  end if
  for a fixed number of iterations **do**
    $i \leftarrow$ Boltzmann Selection $(\mathcal{T}^n)$
    $[j, \mathcal{T}^n] \leftarrow$ Tree Expansion $(\mathcal{T}^n, i, h)$
    $a^n \leftarrow$ Simulation (j, B)
    $a^{-n} \leftarrow$ Sample $(\hat{\mathcal{A}}^{-n}, p^{-n})$
    $r(a^n) \leftarrow$ Marginal Contribution (g, a^n, a^{-n})
    $\mathcal{T}^n \leftarrow$ Backpropagation $(\mathcal{T}^n, j, r(a^n))$
    $t \leftarrow t + 1$
  end for
  $[\hat{\mathcal{A}}^{-n}, p^{-n}] \leftarrow$ Update and Communicate $(\hat{\mathcal{A}}^n, p^n)$
end while
return $a^n \leftarrow \arg\max_{a \in \hat{\mathcal{A}}^n} [p^n(a)]$

To evaluate its rollout a^n, agent n samples joint actions a^{-n} for other agents from $(\hat{\mathcal{A}}^{-n}, p^{-n})$ and computes its marginal contribution:

$$r(a^n) = g(a^n, a^{-n}) - g(a^{-n}), \tag{1}$$

which aligns each agent's objective with the global utility while mitigating variance in multi-agent evaluation (Wolpert, Bieniawski, and Rajnarayan 2013).

In *backpropagation*, discounted updates account for evolving agent intentions. Let N_i be the discounted visit count for node i:

$$N_i = \sum_{t=1}^{T} \gamma^{T-t} \mathbf{1}_{\{a^t = i\}}, \qquad \gamma \in [0.5, 1), \tag{2}$$

with the corresponding discounted value estimate

$$\bar{X}_{i,N_i} = \frac{1}{N_i} \sum_{t=1}^{T} \gamma^{T-t} r^t \mathbf{1}_{\{a^t = i\}}. \tag{3}$$

where r^t is the rollout score at iteration t as defined in (1). After exhausting the computation budget, each agent selects the rollout in $\hat{\mathcal{A}}^n$ with the highest probability under p^n.

Boltzmann Selection Policy

Selection in distributed MCTS is challenging due to non-stationary node statistics: deeper expansions alter reward distributions, and stochastic oversampling of low-value nodes can hinder search efficiency. To address this, CB-MCTS employs a temperature-controlled Boltzmann policy with entropy regularization and decaying uniform exploration. For a node i with children $\mathcal{C}(i)$, the probability of

selecting child j at iteration t is

$$\pi_{i,t}(j) = (1 - \lambda_{i,t})\, \rho_{i,t}(j) + \frac{\lambda_{i,t}}{|\mathcal{C}(i)|}, \tag{4}$$

where $\lambda_{i,t} = \min(1, \epsilon / \log(e + N_i))$ introduces controlled uniform exploration, and

$$\rho_{i,t}(j) \propto \exp\left(\frac{\bar{X}_{j,N_j} + \beta(N_i) H_j}{\alpha(N_i)}\right) \tag{5}$$

is an entropy-regularized Boltzmann distribution. Here $\alpha(\cdot)$ and $\beta(\cdot)$ are decaying schedules, and H_j is the entropy bonus promoting structured early exploration. The entropy is initialized as $H_i = 0$ when the node i is first expanded and dynamically backed up in the backpropagation phase as:

$$H_i \leftarrow \mathcal{H}(\pi_{i,t}) + \sum_{j \in \mathcal{C}(i)} \pi_{i,t}(j)\, H_j, \tag{6}$$

with $\mathcal{H}$ denoting Shannon entropy.

Simple Regret Analysis

CB-MCTS is designed to minimize simple regret by combining structured stochastic exploration with discounted coordination signals. The Boltzmann policy ensures that all actions remain discoverable while gradually concentrating probability mass on high-value branches. Meanwhile, the marginal contribution objective and discounted backups attenuate outdated information, enabling each agent to adapt to the evolving intentions of others. Together, these mechanisms promote consistent alignment between local rollouts and the global objective, allowing the search to converge more rapidly toward high-reward regions.

Theorem 2 *The simple regret of CB-MCTS, with* $\alpha(\cdot) \to 0$ *and* $\beta(\cdot) \to 0$*, is bounded by* $\mathbb{E}[r_T] \leq C \exp(-kT/\log T)$ *for some constants* $C, k > 0$.

Theorem 2 shows that the simple regret of CB-MCTS decays exponentially faster in T than Dec-MCTS with D-UCT. As illustrated by the D-chain problem (Figure 1), the simple regret of CB-MCTS vanishes with far fewer iterations, regardless of the value of γ, indicating that it identifies optimal actions more rapidly. This is especially valuable in applications with limited planning resources. The proof of Theorem 2 and additional simple regret analysis in the D-chain problem are in the extended version (Nguyen et al. 2026).

Empirical Evaluation

We evaluate CB-MCTS on different multi-agent cooperative planning tasks. We consider the following baselines:

- *Dec-MCTS*: leading version of decentralized MCTS with D-UCT (Best et al. 2019).
- *GU-MCTS*: CB-MCTS using global utility instead of marginal contributions.
- *NE-MCTS*: CB-MCTS without entropy ($\beta(m) = 0$).
- *Independent*: CB-MCTS runs independently per agent (equivalent to the single-agent algorithm AR-DENTS (Painter et al. 2023)).

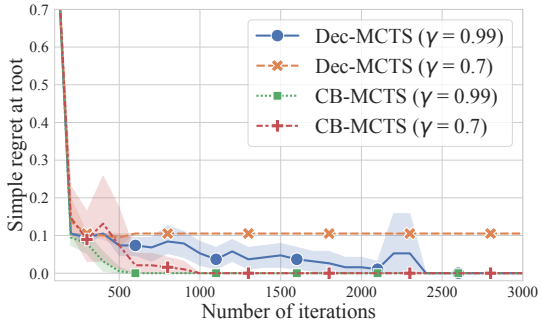

Figure 1: Simple regret of CB-MCTS and Dec-MCTS in the multi-agent D-chain problem with $D = 10$ and 2 agents.

- *CAR-DENTS*: AR-DENTS adapted for centralized multi-agent planning where a single tree encodes all agents, with agent n acting at depths $(n, n + N, n + 2N, \dots)$.

We tuned all methods on small validation instances. The hyperparameter selections and environment setup details are in the extended version (Nguyen et al. 2026).

Frozen Lake Problem

We consider the *Frozen Lake* benchmark (Towers et al. 2024), a grid-world where each cell is either safe or a hole. The agent starts in the top-left corner and moves until reaching a goal, falling into a hole, or exhausting its budget. We extend the task to a multi-agent setting with two goal positions. An agent reaching a goal at step t receives a score of 0.99^t; otherwise it receives 0. Multiple agents selecting the same goal do not yield additional reward. This setup generalizes the multi-goal stochastic navigation problems. Beyond the joint score, we measure the probability that at least one goal is reached (*PR1*) and the probability that both goals are reached (*PR2*). Each algorithm is evaluated over 80 runs on four 8×12 maps with two goals, using a planning budget of 100 steps and reporting metrics every 250 iterations.

As shown in Figure 2, CB-MCTS reaches both goals up to 40% more often than Dec-MCTS and attains a 70% higher joint score. The problem's sparse reward structure favors Boltzmann-based exploration, which increases the chance of discovering successful trajectories. The entropy-guided search further mitigates premature termination by avoiding low-entropy (hole-adjacent) actions. Without this mechanism, NE-MCTS exhibits a substantial performance drop.

Independent and CAR-DENTS can reach at least one goal (Figure 2b) but frequently miscoordinate, sending both agents to the same target. GU-MCTS can eventually match CB-MCTS in joint score, but directly optimizing the global utility yields high-variance estimates and unstable coordination. In contrast, CB-MCTS leverages marginal contributions to decouple each agent's influence, enabling faster convergence. As shown in Figure 2c, CB-MCTS achieves a 60% PR2 level twice as fast and an 80% PR2 level 1.5× faster than GU-MCTS. Overall, the results demonstrate that the components of CB-MCTS collectively provide robust performance on reward-sparse, decentralized planning tasks.

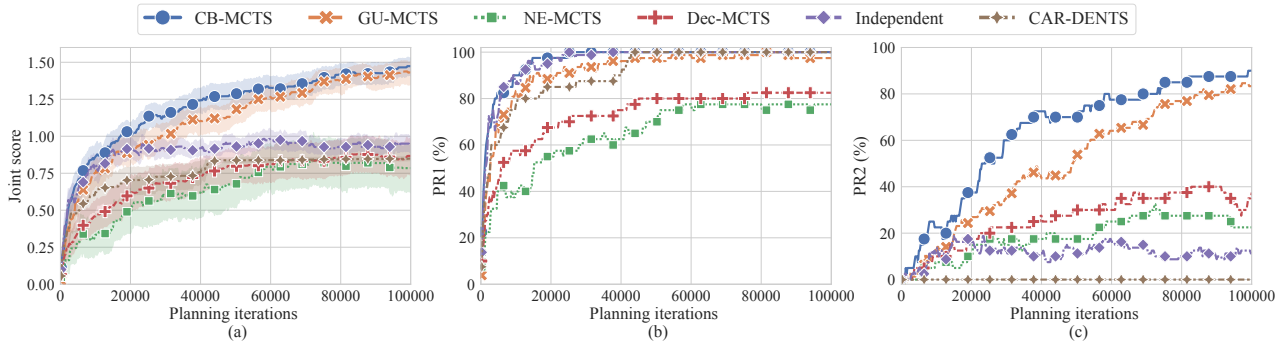

Figure 2: Performance comparison on the Frozen Lake benchmark.

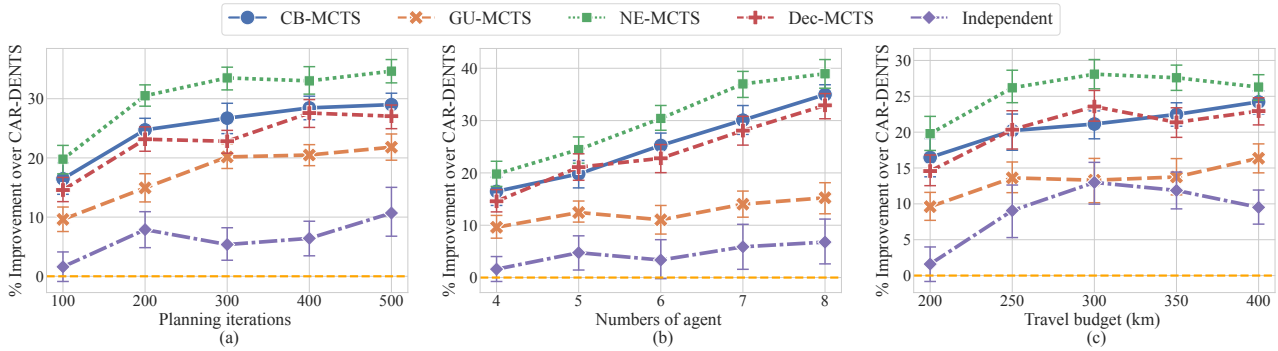

Figure 3: Performance comparison in the Oil Rigs Inspection problem.

Oil Rigs Inspection Problem

We consider an oil rig inspection task in a 200 km $\times$ 100 km region containing 1000 oil rigs (ORs). A team of N autonomous vehicles, starting from a common depot, must visit 200 randomly selected ORs, each with a 2.5 km observation radius. Agents move on an undirected graph G covering the region, where every OR intersects at least one edge. An OR is *visited* when an agent traverses an edge within its observation range. All agents know G and plan paths to maximize total OR coverage under a uniform travel budget. This setup generalizes multi-robot informative path planning problems and is widely used for evaluating decentralized MCTS methods (Best et al. 2019; Nguyen et al. 2022). Each algorithm is run 40 times over 4 OR subsets. Performance is measured as the percentage of visited ORs. CAR-DENTS is evaluated in a centralized training, decentralized execution (CTDE) regime, where all agents' actions are planned offline for 3000 iterations. Distributed algorithms use online replanning: each agent plans, executes its first edge, and replans until exhausting its budget.

Figure 3 summarizes performance under varying parameters (default: 4 agents, 100 planning iterations per cycle, 200 km travel budget). Despite the dense and smooth reward landscape, which typically favors UCT-style planners, CB-MCTS consistently matches Dec-MCTS and surpasses it with additional planning iterations. Dense rewards also increase coordination complexity, since overlapping coverage reduces global value. Accordingly, GU-MCTS (optimizing global utility directly) and Independent perform notably worse due to high-variance value estimates and limited coordination. The online distributed methods further benefit from parallelization, allowing agents to expand deeper local search trees and outperform the CTDE baseline.

Notably, NE-MCTS consistently performs best and maintains a 5–10% improvement over Dec-MCTS. This suggests that in environments with dense, smooth reward distributions, removing entropy can lead to better empirical performance, as the Boltzmann temperature schedule effectively controls exploration. It also improves computational efficiency by lowering search variance and runtime overhead. Taken together, these results show that CB-MCTS is scalable and adaptable to a wide range of multi-agent planning problems, from smooth to sparse reward environments.

Conclusion

Efficient coordination in multi-agent planning remains challenging, especially when optimal actions initially appear suboptimal. We introduced CB-MCTS, a distributed algorithm that promotes early exploration while optimizing collective utility. Experiments show that CB-MCTS matches state-of-the-art methods in general settings and significantly outperforms them in deceptive environments requiring extensive exploration. Future work will explore how adversarial perturbations affect cooperative planning and evaluate the robustness of CB-MCTS under such conditions.

Acknowledgments

This research was supported in part by the UNITY-6G project, co-funded by the European Union's Horizon Europe Smart Networks and Services Joint Undertaking (SNS JU) under Grant Agreement No. 101192650. Additional support was provided by the Swiss State Secretariat for Education, Research and Innovation (SERI), as well as by the FAIR GANDEEP and RADIANT projects.

References

Best, G.; Cliff, O. M.; Patten, T.; Mettu, R. R.; and Fitch, R. 2019. Dec-MCTS: Decentralized planning for multi-robot active perception. *Int. J. Robot. Res.*, 38(2-3): 316–337.

Bubeck, S.; Munos, R.; and Stoltz, G. 2011. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19): 1832–1852.

Cesa-Bianchi, N.; Gentile, C.; Lugosi, G.; and Neu, G. 2017. Boltzmann exploration done right. *NeurIPS*, 30.

Claes, D.; Oliehoek, F.; Baier, H.; Tuyls, K.; et al. 2017. Decentralised online planning for multi-robot warehouse commissioning. In *AAMAS*, 492–500.

Coquelin, P.-A.; and Munos, R. 2007. Bandit algorithms for tree search. In *UAI*, 67–74.

James, S.; Konidaris, G.; and Rosman, B. 2017. An analysis of monte carlo tree search. In *AAAI*, volume 31.

Kocsis, L.; Szepesvári, C.; and Willemson, J. 2006. Improved Monte-Carlo Search. *Univ. Tartu, Estonia, Tech. Rep*, 1.

Li, M.; Yang, W.; Cai, Z.; Yang, S.; and Wang, J. 2019. Integrating Decision Sharing with Prediction in Decentralized Planning for Multi-Agent Coordination under Uncertainty. In *IJCAI*, 450–456.

Munos, R.; et al. 2014. From bandits to Monte-Carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1): 1–129.

Nguyen, N.; Nguyen, D.; Kim, J.; Rizzo, G.; and Nguyen, H. 2022. Multi-Agent Data Collection in Non-Stationary Environments. In *WoWMoM*, 120–129. IEEE.

Nguyen, N.; Nguyen, D.; Kim, J.; Rizzo, G.; and Nguyen, H. 2024a. Decentralized Coordination for Multi-Agent Data Collection in Dynamic Environments. *IEEE Trans on Mobile Computing*.

Nguyen, N.; Nguyen, D.; Rizzo, G.; and Nguyen, H. 2024b. United We Stand: Decentralized Multi-Agent Planning With Attrition. In *ECAI*, 3421–3428. IOS Press.

Nguyen, N.; Nguyen, D.; Rizzo, G.; and Nguyen, H. 2026. Boltzmann-based Exploration for Robust Decentralized Multi-Agent Planning. *arXiv preprint arXiv:2603.02154*.

Painter, M.; Baioumy, M.; Hawes, N.; and Lacerda, B. 2023. Monte Carlo tree search with Boltzmann exploration. *Advances in Neural Information Processing Systems*, 36: 78181–78192.

Ramanujan, R.; and Selman, B. 2011. Trade-offs in sampling-based adversarial planning. In *ICAPS*, volume 21, 202–209.

Sukkar, F.; Best, G.; Yoo, C.; and Fitch, R. 2019. Multi-robot region-of-interest reconstruction with Dec-MCTS. In *ICRA*, 9101–9107. IEEE.

Tolpin, D.; and Shimony, S. 2012. MCTS based on simple regret. In *AAAI*, volume 26, 570–576.

Towers, M.; Kwiatkowski, A.; Terry, J.; Balis, J. U.; De Cola, G.; Deleu, T.; Goulão, M.; Kallinteris, A.; Krimmel, M.; KG, A.; et al. 2024. Gymnasium: A Standard Interface for Reinforcement Learning Environments. *arXiv preprint arXiv:2407.17032*.

Wolpert, D. H.; Bieniawski, S. R.; and Rajnarayan, D. G. 2013. Probability collectives in optimization. *Handbook of Statistics*, 31: 61–99.